

420 Rec'd PCT/PTO 08 NOV 1999

## SPECIFICATION

## PORTABLE VIEWING/LISTENING SYSTEM

## TECHNICAL FIELD

The present invention relates to portable viewing/listening systems, more particularly to a portable viewing/listening system in which program data received by a fixedly or semi-fixedly installed main device is transferred to an easy-to-carry portable device so as to provide a user with the program data played-back on the portable device side.

## BACKGROUND ART

Thanks to multi-channeled broadcast and developing video services through communications, the number of programs available for general users has been increasing. However, the users' spare-time for watching the programs is limited within a day. Accordingly, the users cannot always watch every program interested.

Recently, a portable viewing/listening system which enables the users to watch programs in their spare time (during lunch or in transit) has been proposed. A conventional portable viewing/listening system is constituted by a main device installed in a house and an easy-to-carry portable device. The main device is at least capable of receiving program data provided through broadcasting, internally-storing the received program data, and transferring the stored program data to the portable

device.

The conventional portable viewing/listening system, however, takes long to transfer the program data from the main device to the portable device. Accordingly, if the users run out of time and thus stop data transfer halfway through, the program data is not entirely transferred, and thus the users cannot enjoy the entire program.

FIG. 18 is a diagram illustrating a data transfer mode adopted in the conventional portable viewing/listening system. Hereinafter, by referring to FIG. 18, problems associated with the conventional system are described in more detail.

FIG. 18(a) shows time transition of data constituting a program to be broadcast. In FIG. 18(a), a lateral axis indicates a lapse of time during broadcast, and a vertical axis indicates a bit rate during broadcast, respectively. To be more specific, FIG. 18(a) shows that Program 1 is broadcast between Time  $t_0$  and Time  $t_1$ , Program 2 is broadcast between Time  $t_1$  and Time  $t_2$ , and Program 3 is broadcast between Time  $t_2$  and Time  $t_3$ , respectively. The Program 1, Program 2, and Program 3 are respectively provided with audio, closed caption, and video as program components (mono-media constituting a program). For the sake of clarity, it is herein assumed that a length of broadcast time for the Program 2 ( $t_2 - t_1$ ) is equal to that for the Program 3 ( $t_3 - t_2$ ), and a length of broadcast time for the Program 1 ( $t_1 - t_0$ ) is exactly twice that for the Program 2 (or the Program 3). It is also assumed

that a bit rate for the respective program components remains invariant no matter which Program, and thus a ratio among a bit rate of closed caption  $bc$ , bit rate of audio  $ba$ , and bit rate of video  $bv$  is always 1:2:4.

FIG. 18(b) shows time transition taken place when the program data is transferred from the main device to the portable device in the conventional portable viewing/listening system. In FIG. 18(b), a lateral axis indicates a lapse of time in data transfer operation, and a vertical axis indicates a bit rate during transfer, respectively. As shown in FIG. 18(b), transfer of the program components of the Program 1, Program 2, and Program 3 is completed between Time  $t_4$  and Time  $t_9$ . In this drawing, unlike the bit rate during broadcast shown in FIG. 18(a), the bit rates for the program components during transfer are each compressed or extended in the direction of a time axis according to characteristics of data for the respective program components. Basically, however, the program components herein are transferred in the same order as those during broadcast. It means, every program component of the Program 1 is transferred between Time  $t_4$  and Time  $t_7$ , every program component of the Program 2 is transferred between Time  $t_7$  and Time  $t_8$ , and every program component of the Program 3 is transferred between Time  $t_8$  and Time  $t_9$ , respectively. A ratio among  $(t_7 - t_4)$ ,  $(t_8 - t_7)$  and  $(t_9 - t_8)$ , all of which indicate a length of time taken to transfer each Program, is 2:1:1. This is equal to the ratio among  $(t_1 - t_0)$ ,

( $t_2 - t_1$ ) and ( $t_3 - t_2$ ), all of which indicate a length of broadcast time for each Program. Assuming that a total bit rate for closed caption, audio and video during transfer is  $\alpha$ -time as much as a total bit rate during broadcast, a ratio between a length of broadcast time of a certain program and a length of time taken to transfer the program is always  $\alpha:1$ .

FIG. 18(c) shows a case where transfer of program data is discontinued halfway at Time  $t_a$  in the conventional portable viewing/listening system. In this example, the Time  $t_a$  is assumed to satisfy  $t_7 < t_a < t_8$ . As transfer of the program data is discontinued in the middle of the Program 2 between Time  $t_7$  and Time  $t_8$ , data of the Program 1 in the entirety and data of the Program 2 partially from the beginning to Time  $t_a$  is transferred to the portable device. However, as to the remaining data of the Program 2 and data of the Program 3 in the entirety, none of the program components, i.e., closed caption, audio and video included therein is transferred. As to the not-transferred part of data, the portable device does not tell the users what it is. Therefore, the users cannot even know whether or not the data is interesting enough to watch.

#### DISCLOSURE OF THE INVENTION

Therefore, an object of the present invention is to provide a portable viewing/listening system which guarantees the minimum play-back quality required for the users on the portable device side even when data transfer from the main device to the portable

device is discontinued in a short time.

The present invention has the following features to attain the object above.

A first aspect of the present invention is directed to a portable viewing/listening system in which program data received by a fixedly or semi-fixedly installed main device is played-back in an easy-to-carry portable device to provide a user therewith,

the main device comprising:

receiving means for receiving program data provided through broadcasting;

primary storing means for storing the program data received by the receiving means;

transfer priority assigning means for assigning transfer priorities to every program component constituting the program data stored in the primary storing means; and

transferring means for discretely, to the portable device, each of the program components of the program data stored in the primary storing means in the direction of a time axis in order of the assigned transfer priorities, and

the portable device comprising:

secondary storing means for storing the program components to be discretely transferred from the transferring means;

re-constructing means for re-constructing program

data included in any program desired to play-back from the program components stored in the secondary storing means; and

playing-back means for playing-back the program data re-constructed by the re-constructing means.

As described above, in the first aspect, each program component constituting the program data simultaneously received by the main device is assigned transfer priority, and the program components are discretely transferred, to the portable device, in the direction of the time axis in order of the assigned transfer priorities. Therefore, even if data transfer from the main device to the portable device is discontinued in a short time, there is a high possibility that some of the program components are entirely transferred to the portable device. Accordingly, the users can enjoy, yet incomplete, the entire program by playing-back the already-transferred program components.

According to a second aspect, in the first aspect, the transfer priority assigning means assigns transfer priorities to each of the program components in such a manner that a program component with less amount of data has a higher transfer priority.

As described above, in the second aspect, any program component having less amount of data has a higher transfer priority from the main device to the portable device. It means, a length of time taken to transfer the program component having the higher transfer priority is shorter. Consequently, if data

transfer is presumably discontinued in this system and in the conventional one at the same time, the number of program components successfully transferred to the portable device in this system is larger.

According to a third aspect, in the second aspect, the main device further comprises transfer priority defining means for previously defining transfer priorities by default to every type of the program components, and

the transfer priority assigning means assigns, in accordance with a definition provided by the transfer priority defining means, the transfer priorities by default to each of the program components.

As described above, in the third aspect, each of the program components is assigned, in the main device, the transfer priorities by default according to the type thereof. Therefore, the program components can be transferred in an order appropriately determined by the characteristics thereof.

According to a fourth aspect, in the third aspect, when the plurality of program data stored in the primary storing means is collectively transferred to the portable device, the transferring means selects any program component having the same transfer priority from each of the program data, combines the selected program components into a unit, and then discretely transfers the combined unit in the direction of the time axis in order of the transfer priorities assigned to every combined unit.

As described above, in the fourth aspect, when a plurality of program data is collectively transferred to the portable device, the data is not transferred on a program basis but on a transfer priority basis. That is, the program components having the same transfer priority are selected from the respective program data, the selected program components are combined to a unit, and then the combined unit is transferred in order of the transfer priorities assigned to every combined unit. Therefore, the programs are assigned transfer priorities in a fair manner, and resultantly variation in arriving time of the programs can be prevented.

According to a fifth aspect, in the fourth aspect, types of the program components are classified into a plurality of classes by the transfer priority defining means, and the transfer priority assigning means checks every class for the transfer priorities by default assigned to each of the program components, and then changes, in accordance with a result of the check, the transfer priorities by default assigned to each of the program components on a class basis, as required.

As described above, in the fifth aspect, the types of the program components are classified into a plurality of classes, and the transfer priorities are checked on a class basis before making a change, if any. Therefore, the transfer priorities can be managed on a class basis.

According to a sixth aspect, in the fifth aspect,



when no program component in the same class has a value defined as being top transfer priority therein, the transfer priority assigning means changes a value of a program component being currently the highest transfer priority in the class to the value defined as being top therein.

As described above, in the sixth aspect, even if the types of the program components in the same class are varying, some of the program components are managed to reach the portable device when the first unit for each class is transferred.

According to a seventh aspect, in the first aspect, the main device further comprises program component generating means for generating a new program component from the program components of the program data received by the receiving means, and

the storing means adds the program component newly generated by the program component generating means to the program data received by the receiving means, and stores the same.

As described above, in the seventh aspect, another program component can be newly generated from the received program components and transferred. It is effective for a case where the received program data is hierarchically encoded into a plurality of layers so that the program components in the respective layers are transferred, or a case where the received detailed video is decimated to generate and transfer a non-detailed video.

According to an eighth aspect, in the first aspect,

when any data of the program component of the program data re-constructed by the re-constructing means is discontinued halfway during play-back by the playing-back means, the portable device further comprises replacing means for having any other program component being not currently played-back to play-back as a replacement.

As described above, in the eighth aspect, the program component which was not entirely played-back due to the discontinued data transfer is replaced by any other program component to play-back. Therefore, the users can enjoy the entire program.

According to a ninth aspect, in the eighth aspect, the portable device further comprises presentation priority defining means for previously defining presentation priorities to each type of the program components, and

the replacing means determines a program component to play-back as a replacement in accordance with the definition provided by the presentation priority defining means.

As described above, in the ninth aspect, the portable device is set to determine the program component as a replacement in accordance with the presentation priorities defined to every type of the program components. Therefore, the program components can be replaced to play-back in an order suitably determined by the characteristics thereof.

According to a tenth aspect, in the ninth aspect,

types of the program components are classified into a plurality of classes by the presentation priority defining means, and

the replacing means determines a program component to play-back as a replacement among the program components belonging to the same class as does the program component discontinued during play-back.

As described above, in the tenth aspect, the replacing order to play-back can be managed on a class basis.

According to an eleventh aspect, in the first aspect, the main device and the portable device are structured to be electrically interconnectable, and

the transferring means online-transfers each program component of the program data stored in the primary storing means directly to the portable device.

As described above, in the eleventh aspect, the main device and the portable device are electrically interconnectable. Consequently, program data can be on-line transferred.

According to a twelfth aspect, in the eleventh aspect, the main device further comprises:

attaching means for allowing the portable device to be electrically connected; and

charging means for supplying power to charge the portable device when the portable device is attached to the attaching means, wherein

the portable device further comprises a battery to be charged by the power supplied from the charging.

As described above, in the twelfth aspect, transfer of the program data and charging of the portable device can be done at the same time.

According to a thirteenth aspect, in the first aspect, the transferring means offline-transfers each program component of the program data stored in the primary storing means to the portable device through a recording medium.

As described above, in the thirteenth aspect, data can be off-line transferred from the main device to the portable device through a recording medium. Therefore, such structure eases restrictions imposed on the physical layout of the main device and the portable device. Furthermore, a plurality of recording media will increase a substantial memory capacity.

According to a fourteenth aspect, in the thirteenth aspect, the main device further comprises writing means for writing any program component to be transferred to the portable device into the recording medium, and

the portable device further comprises reading means for reading the program component recorded in the recording medium.

According to a fifteenth aspect, in the first aspect, the receiving means receives program data through a computer network.

A sixteenth aspect of the present invention is directed to

a main device used in a state of fixedly or semi-fixedly installed, and used to transfer any received program data to an easy-to-carry portable device, the main device comprising:

receiving means for receiving program data provided through broadcasting;

primary storing means for storing the program data received by the receiving means;

transfer priority assigning means for assigning transfer priorities to each program component constituting the program data stored in the primary storing means; and

transferring means for discretely, to the portable device, each of the program components of the program data stored in the primary storing means in the direction of a time axis in order of the assigned transfer priorities.

As described above, in the sixteenth aspect, each of the program components constituting program data simultaneously received is assigned transfer priority, and the program components are discretely transferred, to the portable device, in the direction of the time axis in order of the assigned transfer priorities. Therefore, even when data transfer to the portable device is discontinued in a short time, some of the program components may possibly be entirely transferred thereto. Accordingly, the users can enjoy the whole program, yet incomplete, by playing-back the already-transferred program components.

According to a seventeenth aspect, in the sixteenth aspect,

the transfer priority assigning means assigns transfer priorities to each program component in such a manner that a program component with less amount of data has a higher transfer priority.

As described above, in the seventeenth aspect, any program component having less amount of data has the higher transfer priority. It means, a length of time taken to transfer the program component having the higher transfer priority is shorter. Consequently, if data transfer is presumably discontinued in this system and in the conventional one at the same time, the number of program components successfully transferred to the portable device in this system is larger.

According to an eighteenth aspect, in the seventeenth aspect,

the main device further comprises transfer priority defining means for previously defining transfer priorities by default to every type of the program components, and

the transfer priority assigning means assigns the transfer priorities by default to each of the program components in accordance with a definition provided by the transfer priority defining means.

As described above, in the eighteenth aspect, each of the program components is assigned the transfer priorities by default according to the type thereof. Therefore, the program components can be transferred in an order appropriately determined by the

characteristics thereof.

According to a nineteenth aspect, in the eighteenth aspect, when the plurality of program data stored in the primary storing means is collectively transferred to the portable device, the transferring means selects any program component having the same transfer priority from each of the program data, combines the selected program components into a unit, then discretely transfers the combined unit in the direction of the time axis in order of the transfer priorities assigned to every combined unit.

As described above, in the nineteenth aspect, when a plurality of program data is collectively transferred to the portable device, the data is not transferred on a program basis but on a transfer priority basis. That is, the program components having the same transfer priority are selected from the respective program data, the selected program components are combined to a unit, and then the combined unit is transferred in order of the transfer priorities assigned to every combined unit. Therefore, the programs are assigned transfer priorities in a fair manner, and resultantly variation in arriving time of the programs can be prevented.

According to a twentieth aspect, in the nineteenth aspect, types of the program components are classified into a plurality of classes by the transfer priority defining means, and the transfer priority assigning means checks every class for the transfer priorities by default assigned to each of the

program components, and then changes, in accordance with a result of the check, the transfer priorities by default assigned to each of the program components on a class basis, as required.

As described above, in the twentieth aspect, the types of the program components are classified into a plurality of classes, and the transfer priorities are checked on a class basis before making a change, if any. Therefore, the transfer priorities can be managed on a class basis.

According to a twenty-first aspect, in the twentieth aspect,

when no program component in the same class has a value defined as being top transfer priority therein, the transfer priority assigning means changes a value of a program component being currently the highest transfer priority in the class to the value defined as being top therein.

As described above, in the twenty-first aspect, even when the types of the program components in the same class are varying, some of the program components are managed to reach the portable device when the first unit for each class is transferred.

According to a twenty-second aspect, in the sixteenth aspect,

the main device further comprises program component generating means for generating a new program component from the program components of the program data received by the receiving means, wherein



the storing means adds the program component newly generated by the program component generating means to the program data received by the receiving means, and stores the same.

As described above, in the twenty-second aspect, another program component can be newly generated from the received program components and is transferred. It is effective for a case where the received program data is hierarchically encoded into a plurality of layers so that the program components in the respective layers are transferred, or a case where the received detailed video is decimated to generate and transfer a non-detail video.

A twenty-third aspect of the present invention is directed to a portable device for receiving and playing-back program data received by a fixedly or semi-fixedly installed main device to provide a user therewith, the portable device comprising:

secondary storing means for storing each program component of the program data to be discretely transferred from the main device;

re-constructing means for re-constructing program data of a program desired to play-back from the program components stored in the secondary storing means; and

playing-back means for playing-back the program data re-constructed by the re-constructing means.

As described above, in the twenty-third aspect, in a case where the program components of the program data are discretely

transmitted from the main device, the program data included in any program desired to play-back can be re-constructed from the received program components and is played-back.

According to a twenty-fourth aspect, in the twenty-third aspect,

when data of the program component of the program data re-constructed by the re-constructing means is discontinued halfway during play-back by the playing-back means, the portable device further comprises replacing means for having any other program component being not currently played-back to play-back as a replacement.

As described above, in the twenty-fourth aspect, the program component which was not entirely played-back due to the discontinued data transfer is replaced by any other program component to play-back. Therefore, the users can enjoy the entire program.

According to a twenty-fifth aspect, in the twenty-fourth aspect,

the portable device further comprises presentation priority defining means for previously defining presentation priorities to each type of the program components, and

the replacing means determines a program component to play-back as a replacement in accordance with a definition provided by the presentation priority defining means.

As described above, in the twenty-fifth aspect, the

portable device is set to determine the program component as a replacement in accordance with the presentation priorities defined to every type of the program components. Therefore, the program components can be replaced to play-back in an order suitably determined by the characteristics thereof.

According to a twenty-sixth aspect, in the twenty-fifth aspect,

types of the program components are classified into a plurality of classes by the presentation priority defining means, and

the replacing means determines a program component to play-back as a replacement among the program components belonging to the same class as does the program component discontinued during play-back.

As described above, in the twenty-sixth aspect, the replacing order to play-back can be managed on a class basis.

A twenty-seventh aspect of the present invention is directed to a method of transferring program data received by a fixedly or semi-fixedly installed main device to an easy-to-carry portable device, the method comprising the steps of:

- receiving program data provided through broadcasting;
- storing the program data received by the receiving means;
- assigning transfer priorities to each program component constituting the stored program data; and
- discretely transferring, to the portable device, each of

the program components of the stored program data in the direction of a time axis in order of the assigned transfer priorities.

As described above, in the twenty-seventh aspect, each program component constituting the program data simultaneously received by the main device is assigned transfer priority, and the program components are discretely transferred, to the portable device, in the direction of the time axis in order of the assigned transfer priorities. Therefore, even if data transfer from the main device to the portable device is discontinued in a short time, there is a high possibility that some of the program components are entirely transferred to the portable device. Accordingly, the users can enjoy, yet incomplete, the entire program by playing-back the already-transferred program components.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing the structure of a portable viewing/listening system according to a first embodiment of the present invention.

FIG. 2 is a diagram exemplarily showing a storage list stored in a memory 112 in a main device 1 shown in FIG. 1.

FIG. 3 is a flowchart illustrating an operation on the main device side when program data is transferred from the main device 1 to a portable device 2 in the portable viewing/listening system of the first embodiment.

FIG. 4 is a flowchart illustrating a detailed operation of

a subroutine step S103 in the flowchart of FIG. 3.

FIG. 5 is a diagram exemplarily showing a transfer priority table stored in the memory 112 in the main device 1 shown in FIG. 1.

FIG. 6 is a diagram exemplarily showing a receipt list stored in a memory 208 in the portable device 2 shown in FIG. 1.

FIG. 7 is a flowchart illustrating an operation of the portable device 2 when receiving program data transferred from the main device 1 in the portable viewing/listening system of the first embodiment.

FIG. 8 is a flowchart illustrating how the portable device 2 is operated to play-back a program in the portable viewing/listening system of the first embodiment.

FIG. 9 is a diagram exemplarily showing a class table stored in the memory 208 in the portable device 2 shown in FIG. 1.

FIG. 10 is a diagram illustrating how data is transferred in the portable viewing/listening system in FIG. 1.

FIG. 11 is a diagram showing time transition taken place during program transfer in a case where data to be transferred includes a digest video in the portable viewing/listening system in FIG. 1.

FIG. 12 is a diagram exemplarily showing a storage list to be generated when the received data includes a digest video in the main device 1 in FIG. 1.

FIG. 13 is a diagram exemplarily showing a receipt list to

be generated when the data transferred from the main device 1 includes a digest video in the portable device 2 in FIG. 1.

FIG. 14 is an outer perspective diagram showing the physical structure of a portable viewing/listening system according to a second embodiment of the present invention.

FIG. 15 is a block diagram showing the electric structure of the portable viewing/listening system according to the second embodiment of the present invention.

FIG. 16 is an outer perspective diagram showing the physical structure of a portable viewing/listening system according to a third embodiment of the present invention.

FIG. 17 is a block diagram showing the structure of a portable viewing/listening system according to a fourth embodiment of the present invention.

FIG. 18 is a diagram illustrating a data transfer system adopted in a conventional portable viewing/listening system.

#### BEST MODE FOR CARRYING OUT THE INVENTION

(Outline of this system)

The portable viewing/listening system described in the following embodiments is constituted by a main device fixedly or semi-fixedly installed in a house and an easy-to-carry portable playing-back device (hereinafter, referred to as portable device). The main device includes at least a function of receiving (in some cases, a function of playing-back included) program data provided through broadcasting (both broadcast transmitted over radio waves,

and broadcast transmitted through a cable transmission path included), a function of internally storing the received program data, and a function of online-transferring or offline-transferring the stored program data to the portable device. To online-transfer any selected program data, the portable device is connected to the main device. The portable device then internally stores the program data transferred from the main device, and plays-back the stored program data at users' convenience. To offline-transfer the program data, on the other hand, the main device once writes the internally stored program data to a recording medium (video tape, rewritable optical disk, solid memory card, and the like). After the writing, the recording medium is detached from the main device and then attached to the portable device. Accordingly, in this case, the portable device reads any desirable program data from the attached recording medium and then plays-back the same.

A distinctive characteristic of this system lies in a method of transferring data from the main device to the portable device. Specifically, this system discretely transfers data from the main device to the portable device. As described in the foregoing, one program data is generally constituted by a plurality of program components (video, audio, closed caption, for example). As being concurrently played-back, these program components are conventionally transferred to the portable device at one time. On the other hand, in this system, every program component

constituting one program data is assigned transfer priority so as to be discretely transferred in the direction of a time axis according to the given order of priorities.

With such data transfer discretely executed, some of the program components may possibly be entirely transferred to the portable device even if data transfer from the main device to the portable device is discontinued in a short time. Therefore, the users can enjoy the entire program, yet incomplete, by playing-back the already-transferred program components. In the aforementioned conventional portable viewing/listening system, however, none of the program components of the program data is entirely transferred if data transfer to the portable device is discontinued in a short time. As a result, the users cannot enjoy the entire program.

(First Embodiment)

FIG. 1 is a block diagram showing the structure of a portable viewing/listening system according to a first embodiment of the present invention. In FIG. 1, the portable viewing/listening system of this embodiment includes a main device 1 and a portable device 2. The portable device 2 is detachable/attachable from/to the main device 1.

The main device 1 includes an antenna 101, a receiving part 102, a demultiplexing part 103, an additional information generating part 104, a storage information generating part 105, a storage managing part 106, a storage part 107, a connection



interface 109, a user input-output part 110, a decoder 111, a memory 112, and a CPU 113.

The receiving part 102 includes a tuner and a demodulator which are constituents of a typical digital broadcast receiver. The receiving part 102 receives broadcast wave received by the antenna 101, and outputs a digital stream. Into the digital stream, video information, audio information, additional information and the like are multiplexed. A specific format for the digital stream may be MPEG-2 TS (Transport Stream), for example. Note that MPEG-2 TS, and multiplexing by MPEG-2 TS are standardized by the international standard of MPEG-2 Systems.

The demultiplexing part 103 is constituted by a TS processor, for example, and separates the digital stream outputted from the receiving part 102 into the video information, the audio information, the additional information, and the like. When the digital stream is MPEG-2 TS, the demultiplexing part 103 classifies TS packets being the smallest unit of the digital stream according to values of packet ID (pid) written in TS packet headers so as to separate the multiplexed video information, audio information, additional information, and the like.

The additional information generating part 104 receives information relevant to the additional information among the information included in the digital stream separated by the demultiplexing part 103 so as to generate additional information. A format for the additional information is a section format

defined in DVB-SI being the international standard, for example. In DVB-SI, a method of transmitting information relevant to a broadcast channel (pid, for example) and information relevant to an EPG (Electric Program Guide) is defined as a data format based on the section format. The information relevant to the EPG, for example, is transmitted in the section format called EIT (Event Information Table). To transmit information in the section format, information to be transmitted is divided into a certain size or smaller, and is repeatedly transmitted. In this manner, the additional information generating part 104 collects and merges the divided information into one data, and updates the information if the contents thereof has been changed.

The storage information generating part 105 converts the video information and audio information inputted from the demultiplexing part 103 into a format suitable for storing in the storage part 107, and then outputs the converted information to the storage managing part 106. The storage information generating part 105 also generates both detailed video information and non-detailed video information in accordance with the video information inputted from the demultiplexing part 103 as is required. The non-detailed video information is of inferior image to the detailed video information but has less amount of data. Accordingly, the non-detailed video information is done with a transfer processing quicker than the detailed video information.

Both detailed video information and non-detailed video information are generated in the storage information generating part 105 in a first case where the CPU 113 instructs the storage information generating part 105 to generate the non-detailed video information. If this is the case, the storage information generating part 105 outputs the video information inputted from the demultiplexing part 103 as detailed video information as it is. Further, the storage information generating part 105 changes resolution of the video information inputted from the demultiplexing part 103 or intermittently decimates frames thereof so as to generate and output the non-detailed video information.

Both detailed video information and non-detailed video information are generated in the storage information generating part 105 in a second case where the video information inputted from the demultiplexing part 103 includes the non-detailed video information as well as the detailed video information from the beginning. There may be a case where the received video information is hierarchically encoded or a case where the received video information is transmitted in a hierarchical transmission mode, for example. If this is the case, the video information inputted from the demultiplexing part 103 is multi-layered in varying image qualities, and the upper layer has the more codes and the better image. In this manner, the storage information generating part 105 outputs the video information included in the

top layer as detailed video information without any change. Further, the storage information generating part 105 outputs the video information included in the lower layer as non-detailed video information. To generate video information corresponding to the respective layers, a method of selecting layers used in the typical digital broadcast receiver can be used as it is.

The storage information generating part 105 outputs the aforementioned non-detailed video information to the storage managing part 106 as video information for a digest video. Accordingly, the CPU 113 manages the non-detailed information stored in the storage part 107 as the video information for the digest video. Note, in this embodiment, that the detailed information and non-detailed information are generated only for video. However, the detailed information and non-detailed information may be generated for audio as well depending on cases.

The storage managing part 106 and the storage part 107 are interconnected, and the storage managing part 106 controls information writing and reading with respect to the storage part 107 under the control of the CPU 113. The storage part 107 is constituted by a relatively large-capacity storage device using hard disk or DVD-RAM, for example. The storage part 107 stores program data.

The connection interface 109 is used to physically and electrically connect the main device 1 to other devices, and to transmit video, audio or other digital information from/to the

other devices. Such connection interface 109 may be a typical input-output interface for digital information including an interface of IEEE1394.

The user input-output part 110 receives inputs from the users or presents the users in which state the device is. Such user input-output part 110 is exemplarily constituted by an infrared remote controller, remote control photoreceiving part, pilot lamp, and the like. Information is presented to the users also through an GUI operation utilizing OSD (On Screen Display) included in the video information outputted from the decoder 111. In this case, the users push buttons provided for the remote controller while observing the OSD, for example, so as to play-back or erase stored programs, or preselect or transfer programs.

The memory 112 stores various information relating to the operation of the main device 1. To be more specific, the memory 112 stores program information used to execute operation control of the main device 1, additional information outputted from the additional information generating part 104, storage management information used to manage the information stored in the storage part 107, for example.

The CPU 113 and other constituents provided in the main device 1 are interconnected, and the CPU 113 executes the program information stored in the memory 112 so as to collectively control those constituents.

The portable device 2 includes a connection interface 201, a portable viewing/listening information managing part 202, a storage part 203, a decoder 204, a display 205, a speaker 206, a user input-output part 207, a memory 208, and a CPU 209.

The connection interface 201 is physically and electrically connected to the connection interface 109 in the main device 1, and is used to receive video, audio and other digital information from the main device 1. Such connection interface 201 may be a typical input-output interface for digital information including the interface of IEEE1394.

The portable viewing/listening information managing part 202 and the storage part 203 are interconnected. The storage part 203 is constituted by a storage device using hard disk, DVD-RAM, or solid memory, for example. As is installed in the portable device 2, the storage part 203 is preferred to be downsized. The storage part 203 stores program data to be transferred from the main device 1.

The decoder 204 receives and decodes the program data stored in the storage part 203 so as to convert the data into analog video/audio signals. The display 205 constituted by a liquid crystal display or a small-sized CRT display, for example, receives the analog video signals outputted from the decoder 204 so as to display the corresponding image thereon. The speaker 206 receives the analog audio signals outputted from the decoder 204 so as to output the corresponding audio therefrom.

The user input-output part 207 receives inputs from the users, and presents the users in which state the device is. Such user input-output part 110 is exemplarily constituted by an infrared remote controller, remote control photoreceiving part, pilot lamp, and the like. Information is presented to the users also through the GUI operation utilizing OSD displayed on the display 205. In this case, the users push buttons provided for the remote controller while observing the OSD, for example, so as to set to play-back or erase stored programs, or preselect or transfer programs.

The memory 208 stores various information relevant to an operation of the portable device 2. In detail, the memory 208 stores program information used to execute operation control of the portable device 2, storage management information used to manage the information stored in the storage part 203, and the like.

The CPU 209 and other constituents in the portable device 2 are interconnected, and the CPU 209 executes the program information stored in the memory 208 so as to collectively control these constituents.

FIG. 2 is a diagram exemplarily showing the structure of a storage list to be stored in the memory 112 of the main device 1. This storage list is data for managing the program data stored in the storage part 107 of the main device 1, and is table-structured. Specifically, each line of the table represents a

single program component for a particular program. The storage list of FIG. 2 exemplarily shows that the storage part 107 stores eight program components in total. Each line includes four attributes of event\_id, type, size, and transfer priority.

The attribute "event\_id" is attribute information for identifying the programs. In this example, for the sake of simplicity, event\_id for the Program 1, Program 2 and Program 3 are assumed to be 0x0001, 0x0002, 0x0003, respectively. Herein, "0x" indicates that a hexadecimal digit follows.

The attribute "type" is attribute information for identifying the types of program components. Characters "C", "A", and "V" found in the storage list of FIG. 2 respectively represent Closed Caption, Audio and Video. Digest video represented by "D" is further included therein. This will be described later.

The attribute "size" indicates a size of area in the storage part 107 occupied by the corresponding program component. A size unit herein may simply be a bite, or a block represented by the number of blocks in a given size.

The attribute "transfer priority" is attribute information used to determine in which order the program components are transferred when the program data is transferred from the main device 1 to the portable device 2. In this example, the smaller value has the higher priority.

Among the attributes in FIG. 2, when the program is received, event\_id, type and size can be obtained from the information



transmitted as additional information for the program. The transfer priorities are established when the transfer processing of program is executed.

FIG. 3 is a flowchart illustrating an operation executed on the main device side when the program data is transferred from the main device 1 to the portable device 2 in the portable viewing/listening system of the first embodiment. The processing of FIG. 3 is realized when the CPU 113 in the main device 1 executes the program information stored in the memory 112 in the main device 1. Hereinafter, by referring to FIG. 3, the operation whereby the main device 1 transfers the program data to the portable device 2 is described.

First, the CPU 113 selects a first program among the programs stored in the storage part 107 (step S101). The storage list (see FIG. 2) stored in the memory 112 is referred to know what programs are stored in the storage part 107. Taking the storage list of FIG. 2 as an example, "Program 1" is the one to be selected first from the stored data of the Program 1, Program 2, and program 3. Thereafter, the CPU 113 judges whether or not any program was selected in step S101 (step S102). In this example, as "Program 1" was selected in step S101, the CPU 113 goes to step S103.

In step S103, the CPU 113 sets transfer priorities for each program component constituting the program selected in step S101. For details of this subroutine step S103, see FIG. 4.

By referring to FIG. 4, the CPU 113 sequentially selects the program components constituting the target program (first "Program 1") selected in step S101 (step S201). Thereafter, the CPU 113 judges whether or not any program component was selected in step S201 (step S202). If yes, the CPU 113 provides the program component selected in step S201 with an applicable default value predetermined for every type of program components as transfer priority (step S203).

FIG. 5 shows a transfer priority table in which the default values indicating transfer priorities for every type of the program components are defined. The transfer priority table of FIG. 5 is stored in the memory 112, for example. In FIG. 5, as an example, the default values of 1, 2, 3 and 4 as transfer priorities are respectively assigned to the types C, A, D and V. The transfer priority table of Fig. 5 has a class defined for every type. The class herein shows the classified types of program components. Preferably, data replaceable with each other is classified into the same class. For example, audio can be replaced by closed caption, and detailed video can be replaced by digest video. In step S203, the CPU 113 first refers to the transfer priority table of FIG. 5, then sets transfer priorities by default for every program component in the storage list of FIG. 2.

After step S203 being through, the CPU 113 returns to step S201 and selects the next program component for the target program.

Thereafter, the CPU 113 assigns transfer priority for the selected program component by referring to the transfer priority table of FIG. 5 (step S203).

After steps S201 to S203 being repeated, and after every program component constituting the target program being assigned transfer priority, the program components to be selected in step S201 would have all gone by then. Therefore, the CPU 113 determines that no program component was selected in step S202, and advances to step S204.

In step S204, the CPU 113 selects the classes defined in the transfer priority table of FIG. 5 in the given order. In this example, two classes of "script" and "display" are defined in the transfer priority table of FIG. 5. Accordingly, in step S204, either one of the "script" or "display" is initially selected, secondary the other, and tertiary none. Thereafter, the CPU 113 judges whether or not any class was selected in step S204 (step S205). If yes, the CPU 113 then judges whether or not the program component defined in the selected class as being a top transfer priority is included in the target program (step S206). By referring to the transfer priority table in FIG. 5, for example, one of the program components classified in the class of "script" is in type "C" with transfer priority of "1", and the other is in type "A" with transfer priority of "2". It means, in the class thereof, the program component in type "C" is defined as being the top priority. Similarly, as to the program components

classified in the class of "display", the program component in type "D" is assigned with transfer priority of "3", and thus is defined as being the top priority in the class thereof. Therefore, according to the definition of the transfer priority table of FIG. 5, in step S206, the CPU 113 judges whether or not the target program includes any program component in type "C" when the currently selected class is "script". On the other hand, when the currently selected class is "display", the CPU 113 judges whether or not the target program includes any program component in type "D".

When the target program has no program component defined as being the top transfer priority in the selected class, the CPU 113 selects any program component having the highest transfer priority next to the top, and updates the value thereof to a value defined as being the top priority in the class (step S207). In a case where the class of "script" is currently selected and every program component included in the target program is in type "A", the transfer priority for the program component in type "A" is updated from the default value of "2" to "1", which is defined as being the top priority in the class. Similarly, in a case where the class of "display" is currently selected and every program component included in the target program is in type "V", the transfer priority for the program component in type "V" is updated from the default value of "4" to "3", which is defined as being the top priority in the class. Every update in step S207 is

reflected in the storage list of FIG. 2. That is, any applicable transfer priority for the program component in the storage list is updated. Thereafter, the CPU 113 returns to step S204. Note that, when the target program has any program component defined as being the top transfer priority in the selected class, the CPU 113 returns to step S204 without updating any transfer priority.

Next, the CPU 113 selects the next class from the transfer priority table of FIG. 5, refers to the transfer priority defined for the respective types in the selected class, and judges whether or not transfer priorities by default for every program component included in the target program should be updated. If yes, the CPU 113 updates the applicable values to the predetermined ones.

After steps S205 to S207 being repeated, and after the processing is done with every class, the classes to be selected in step S204 would have all gone by then. Thus, the CPU 113 determines that no class was selected in step S205, terminates the subroutine processing in FIG. 4, then returns to the main routine processing in FIG. 3.

Referring back to FIG. 3, the CPU 113 selects the next program from the storage list of FIG. 2 (step S101), and assigns transfer priorities for every program component constituting the selected program (step S103). After steps S101 to S103 being repeated, and after every program in the storage list being assigned transfer priority, the CPU 113 judges that there is no program to select next (step S102), then advances to step S104.

In step S104, the CPU 113 initially sets a count value N of a counter which indicates transfer priority (hereinafter, refers to as transfer priority counter) to a value of "1" indicating the top transfer priority (step S104). Then, the CPU 113 judges whether or not the count value N of the transfer priority counter is 5 (step S105). Herein, four values of 1, 2, 3 and 4 are available as transfer priorities in the transfer priority table in FIG. 5. Therefore, if  $N = 5$ , it means that every transfer priority is completely assigned. As  $N = 1$  at the beginning, the CPU 113 determines that  $N \neq 5$ , then advances to step S106.

In step S106, the CPU 113 selects one program component whose transfer priority is N (at the beginning,  $N = 1$ ) from the storage list of FIG. 2. Thereafter, the CPU 113 judges whether or not any program component was selected in step S106 (step S107). If yes, the CPU 113 takes out the attribute information of "event\_id", "type", and "size" from the line of the storage list corresponding to the program component selected in step S106, then transfers the same to the portable device 2 (step S108). Then, the CPU 113 takes out main data of the program component selected in step S106, and transmits the same to the portable device 2 (step S109). Thereafter, the CPU 113 returns to step S106, selects another program component whose transfer priority is N from the storage list of FIG. 2, then executes the transfer processing for the selected program component.

After steps S106 to S109 being repeated, and after every

program component having the transfer priority of N being completely transferred, the CPU 113 judges that all the program components to be selected in step S106 are gone (step S107), increments the count value N of the transfer priority counter by 1 (step S110), and then returns to step S105. Then, the CPU 113 executes the transfer processing for the program component having the transfer priority of N (N = 2) after being updated.

After steps S105 to S109 being repeated, and after the transfer processing is done with every program component having the transfer priorities of 1 to 4, N = 5. Thus, the CPU 113 judges that the transfer processing is completely done in step S105, then terminates the processing.

FIG. 6 shows the structure of a receipt list to be stored in the memory 208 in the portable device 2. The receipt list is data used to manage how the data stored in the storage part 203 in the portable device 2 (that is, program data transmitted from the main device 1) is stored, and is table-structured. The lines of the table are in a one-to-one correspondence with the program components stored in the storage part 203. Also, each line of the table has six attributes including receipt order, offset, size, event\_id, type, and length of play-back time.

The attribute "receipt order" indicates the order in which the portable device 2 receives the data of the program components from the main device 1, and a value does not appear twice in the receipt list. Thus, the receipt order can be used as a key for

specifying lines of the receipt list.

The attribute "offset" indicates where the program components being in the one-to-one correspondence with the lines of the receipt list are located in the data received from the main device 1. When transfer of the entire program from the main device 1 to the portable device 2 is presumably considered a bit stream or a file, the attribute "offset" represents the number of bites from the beginning of the transfer. The value of the offset can be represented simply in a bite or the number of blocks in a given size similarly to the case where the attribute "size" in the storage list of FIG. 2 matters.

The attribute "size" represents a size of area in the storage part 203 of the portable device 2 occupied by the program components being in the one-to-one correspondence with the lines of the receipt list. The attribute "size" is also represented in a similar unit to the attribute "offset".

The attribute "event\_id" is attribute information used to identify to which program the program components being in the one-to-one correspondence with the lines of the receipt list respectively belong.

The attribute "type" is attribute information used to identify the types of the program components being in the one-to-one correspondence with the lines of the receipt list.

The attribute "length of play-back time" represents the length of play-back time when the program components being in the



one-to-one correspondence with the lines of the receipt lines are played-back under the same conditions. When data transfer is interrupted halfway through for some reasons, the number of bites of the transferred data is less than that of the same program component which is broadcast. In such case, the length of play-back time is assigned a length of time corresponding to the transferred data.

FIG. 7 is a flowchart illustrating how the portable device 2 is operated to receive program data transmitted from the main device 1 in the portable viewing/listening system of the first embodiment. The processing in FIG. 7 can be realized when the CPU 209 in the portable device 2 executes program information stored in the memory 208 in the portable device 2. Through the processing, the program data transmitted from the main device 1 is stored in the portable device 2, and also the receipt list of FIG. 6 is generated. Hereinafter, referring to FIG. 7, it is described how the portable device 2 is operated to receive the program data transmitted from the main device 1.

First of all, the CPU 209 initializes the receipt list of FIG. 6 (step S301). The easiest way of initialization is done by deleting every line thereof, at the same time, deleting the program data stored in the storage part 203 in the portable device 2. Thereafter, the CPU 209 assigns a value of 0 to a variable `current_offset`, and a value of 1 to a variable `send_count` (step S302). After that, the CPU 209 receives the attribute information

of "size", "event\_id", and "type" to be transmitted from the main device 1 (step S303). Such attribute information is the one transmitted in step S108 in FIG. 3. The CPU 209 then judges whether or not there is any data to receive in step S303 (step S304). If not, the CPU 209 terminates the operation, but if yes, then judges whether or not an available region of the storage part 203 is sufficiently large for the value of the "size" obtained in step S303 (step S305). If yes, the CPU 209 goes to step S306, but if not, returns to step S303.

In step S306, the CPU 209 adds a line to the receipt list of FIG. 6, and then assigns values obtained in step S303 to each corresponding attribute "size", "event\_id", or "type" in the added line. Thereafter, the CPU 209 assigns the value of the variable current\_offset to the attribute "offset" in the added line (step S307). The CPU 209 adds the value of "size" obtained in step S303 to the variable current\_offset (step S208), and then assigns the value of the variable send\_count to the attribute "receipt order" in the added line (step S309). The CPU 209 then increments the value of the variable send\_count by 1 (step S310). After that, the CPU 209 receives the main data of the program components transmitted from the main device 1, and then stores the same in the storage part 203 (step S311). Then, the CPU 209 sets the attribute "length of play-back time" (step S312). Herein, it is assumed that the main data of the program components itself includes, as additional information, a bit rate value indicating

the number of data bit per unit presentation time. In this case, to convert the value of the "size" obtained in step S303 to the length of play-back time, the value of the "size" is divided by the bit rate obtained in the aforementioned manner. Note that, the value of the length of program time can be notified directly from the main device 1 as data as is the case with the "size", for example. Next, the CPU 209 judges whether or not receipt of program data is discontinued halfway through for some reasons (step S313). If yes, the CPU 209 advances to step S314, but other than that, returns to step S303. Such discontinuation may include a case, for example, where the users abruptly disconnect the portable device 2 from the main device 1, and vice versa.

In step S314, the CPU 209 finds a value of the attribute "size" in the receipt list from the size of the program component actually stored in the storage part 203, and resets the value (step S314). Then, the CPU 209 obtains, in a similar manner to step S312, a value for the length of play-back time from the value of the attribute "size" in the receipt list, and then resets the "length of play-back time" in the receipt list (step S315). Then, the CPU 209 terminates the processing for receiving.

FIG. 8 is a flowchart illustrating how the portable device 2 is operated to play-back the program. This processing can be realized when the CPU 209 executes the program information stored in the memory 208 in the portable device 2. Hereinafter, by referring to FIG. 8, it is described how the portable device 2

is operated to play-back the program.

First, a description of a class table used to play-back the program is as follows. FIG. 9 is an exemplary class table. The class table is, for example, stored in the memory 208, and the CPU 209 is at liberty to access thereto. As shown in FIG. 9, lines of the class table are in a one-to-one correspondence with the types of the program components, and each has three attributes of "type", "class", and "presentation priority".

In FIG. 9, the attribute "class" shows the classified types of the program components. By taking this class table as an example, two classes of "script" and "display" are defined therein. The first and second lines thereof show that closed caption and audio are instances belonging to the same class of "script", and the third and fourth lines thereof show that digest video and detailed video are instances belonging to the same class of "display". The attribute "presentation priority" defines in which order the program components in varying types classified into the same class should be presented to the users. In this example, the smaller value has the higher presentation priority. Such class table is referred to to search for some other program components for replacement when the program component being played-back is interrupted halfway.

By referring to FIG. 8, the CPU 209 first executes displaying and selecting operation of the programs (step S401). To be more specific, the CPU 209 generates a list of the programs

stored in the storage part 203 by referring to the receipt list of FIG. 6, and then displays the generated list on the display 205. Thereafter, the CPU 401 let the users select any program desired to play-back from the displayed list. The selection of the programs is done through the user input-output part 207. When it is desired for the list of the programs displayed on the display 205 to include titles of the programs as well, data indicating the title-to-"event\_id" relationship may be sent along with the titles during transmission of the program data from the main device 1 to the portable device 2. For such data indicating the relationship, additional information to be transmitted through broadcasting for the EPG (Electric Program Guide) may be used. Next, the CPU 209 extracts, from the receipt list, every program component of the program selected in step S401 to be played-back (step S402). Thereafter, the CPU 209 generates a list of the program components constituting the selected program according to the extraction result in step S402, and then displays the generated list on the display 205 (step S403). In response thereto, the users select one or more program component(s) to be played-back from the displayed list. The selection of the program components is done through the user input-output part 207. After that, the CPU 209 makes a preparation for concurrent play-back of all the program components selected in step S403 (step S404). By concurrently playing-back the program components, lip synch can be attained, for example, which is synchronization between

a speaker's lip movements in the video and audio.

Thereafter, the CPU 209 executes a play-back processing of the program components prepared in step S404 (step S405). The CPU 209 then judges whether or not any play-back position of the program components being played-back has reached to a tail of the data (step S406). There may be a case where the data is completely played-back, or a case where the data is interrupted. If none of the play-back positions of the program components has reached to the tail, the CPU 209 returns to step S405 so as to continue the play-back processing. While, if any play-back position of the program components has reached to the tail, the CPU advances to step S407.

In step S407, the CPU 209 selects any other program component in the same class as the program component whose play-back position has reached to the data tail but in different type in the order of presentation priorities defined in the class table of FIG. 9. To be more specific, the CPU 209 first extracts every program component constituting the program being played-back from the receipt list of FIG. 6, and then selects, among the extracted program components, any program component belonging to the same class as the program component whose play-back position is judged as reached to the data tail in step S406. Thereafter, the CPU 209 takes out any program component, among the selected program components, whose presentation priority is lower than the one whose play-back position has

reached to the data tail, and then selects the type of the program component whose presentation priority is the highest among the taken-out program components. When the play-back position of the program component in type "V" presumably reaches to the data tail during play-back, the CPU 209 selects the type "D", which is in the same class of "display" as the type "V", and has the presentation priority lower than the type "V" but the highest among the rest.

Thereafter, the CPU 209 judges whether or not any type is selected in step S407 (step S408). If yes, the CPU 209 then judges whether or not the program component in the type selected in step S407 is already being played-back (step S409). This embodiment takes a structure where the users can simultaneously select, in step S403, a plurality of program components in varying types classified into the same class. Therefore, the program component in the type selected in step S407 may possibly be already being played-back. For example, if the play-back position of the program component in type "V" reaches to the data tail when the program components in type "D" and "V" are concurrently played-back, the data of the program component in type "D" is selected in step S407 as described in the foregoing. In such case, the CPU 209 returns to step S405 so as not to repeatedly play-back the program component in the same type, and thus does not replace the interrupted program component with others to play-back. While, if the program component in the type selected in step S407

is not yet played-back, the CPU 209 replaces the interrupted program component with the program component in the selected type to play-back (step S410). In this case, the CPU 209 does not start playing-back from the beginning of the program component in the type selected in step S407, but from the continued part of the interrupted program component. In this example, closed caption and audio are in the same class. Even if the data of audio is interrupted, closed caption thus may take the place by playing-back from the interrupted part. In this manner, the users can enjoy the program not with audio but with closed captions. Thereafter, the CPU 209 returns to step S405.

On the other hand, when no type was selected in step S408, the CPU 209 determines that there is no more component to replace to play-back, and then terminates the play-back processing for the class (step S411). It may happen when the users have selected the type having the lowest presentation priority. Also, it may happen when the target program is completely played-back. At this time, the CPU 209 changes a value of a flag variable indicating the play-back processing being executed on the class basis into a value indicating the processing being terminated. Then, the CPU 209 judges whether or not the program component being played-back is completely played-back. This judgement is made by referring to the value of the aforementioned flag variable. If any program component is still being played-back in whatever the class, the CPU 209 returns to step S405 so as to continue the



play-back processing for other classes. While, if the target program is completely played-back, the program components being played-back in every class are terminated. Thus, the CPU 209 terminates the play-back processing of the program.

FIG. 10 is a diagram illustrating how data is transferred in the portable viewing/listening system of FIG. 1. Hereinafter, it is described in detail how data is transferred in the portable viewing/listening system of FIG. 1.

FIG. 10 (a) shows time transition of data constituting the program to be broadcast. In FIG. 10 (a), a lateral axis indicates a lapse of time during broadcast, and a vertical axis indicates a bit rate during broadcast, respectively. In detail, FIG. 10 (a) shows that the Program 1 is broadcast between Time  $t_0$  and Time  $t_1$ , the Program 2 is broadcast between Time  $t_1$  and Time  $t_2$ , and the Program 3 is broadcast between Time  $t_2$  and Time  $t_3$ , respectively. Further, the Program 1, Program 2, and Program 3 are respectively provided with audio, closed caption, and video as a program component (mono-media constituting the program). For the sake of simplicity, the length of broadcast time for the Program 2 ( $t_2 - t_1$ ) and the length of broadcast time for the Program 3 ( $t_3 - t_2$ ) are equal, and the length of broadcast time of the Program 1 ( $t_1 - t_0$ ) is exactly twice that for the Program 2 (or Program 3). Further, a bit rate for the respective program components remains invariant no matter which Program, and thus a ratio among a bit rate of closed caption  $bc$ , bit rate of audio

ba, and bit rate of video bv is always 1:2:4.

FIG. 10(b) shows time transition taken place when program data is transferred from the main device 1 to the portable device 2. In FIG. 10(b), a lateral axis indicates a lapse of time for a data transfer operation, and a vertical axis indicates a bit rate during transfer, respectively. It is assumed that transfer is started at Time t4 as is in the conventional system shown in FIG. 10(b). As the total number of bits to be transferred is equal to the example in FIG. 8, the data is completely transferred at Time t9 as is in the conventional system. In FIG. 10(b), every closed caption part of the programs to be transferred (Program 1, Program 2, and Program 3) is first transferred between Time 4 to Time t5, secondly every audio part thereof is transferred between Time t5 to Time t6, and lastly every video part thereof is transferred between Time t6 to Time t9. As is described in the foregoing, the rate among the program components of closed caption, audio, and video during broadcast is 1:2:4. Therefore, the length of time taken to transfer closed caption of the programs (t5 - t4), the length of time taken to transfer audio thereof (t6 - t5), and the length of time taken to transfer video thereof (t9 - t6) are respectively expressed by the following equations (1), (2) and (3).

$$(t5 - t4) = (t9 - t4) \\ \times \{1 / (1 + 2 + 4)\} \quad \dots (1)$$

$$(t6 - t5) = (t9 - t4)$$

$$\times \{2 / (1 + 2 + 4)\} \quad \dots (2)$$

$$(t9 - t6) = (t9 - t4)$$

$$\times \{4 / (1 + 2 + 4)\} \quad \dots (3)$$

Further,  $t6$  can be represented by the following equation (4).

$$t6 = t4 + [(t9 - t4) \times \{(1 + 2) / (1 + 2 + 4)\}] \quad \dots (4)$$

FIG. 10(c) shows, in the portable viewing/listening system in FIG. 1, a case where transfer of program data is interrupted at Time  $t_a$ . In this example, Time  $t_a$  presumably satisfies  $t6 < t_a < t9$ . Even if transfer of the program data is discontinued halfway through at Time  $t_a$ , audio components and closed caption components for the respective programs are completely transferred by then. At this stage, in the portable device, the programs are not available in video but in closed caption and audio. It means, the users can be informed of the contents thereof to some extent. In other words, the users can enjoy every program with closed caption and audio, yet with no video. In practical sense, the bit rates of the closed caption and audio are rather high compared to that of video. In this sense, Time  $t6$  when every closed caption component and audio component are completely transferred possibly comes at much earlier timing than Time 9 when the program is entirely transferred with the video components. In detail, the length of time required for transferring the video ( $t9 - t6$ ) is

rather long compared to that for the closed caption components and the audio components ( $t_6 - t_4$ ). Therefore, the audio components may highly-possibly be entirely transferred even if transfer is stopped in a rather short time. Further, as is transferred in text, the bit rate of the closed caption components can be lowered a greater extent than that of the audio components. Therefore, transfer of the closed caption components can be completely done much quicker than that of the audio components.

FIG. 11 shows time transition taken place to transfer the program in a case where data to be transferred includes a digest video in the portable viewing/listening system in FIG. 1. The case herein is somewhat complicated compared to the case in FIG. 10.

FIG. 11 (a) shows time transition for information of the program to be broadcast. This example herein is different from the example in FIG. 10(a) that the Program 1 includes a digest video D1 in a detailed video V1, and the Program 2 does not include a closed caption C2. The bit rate without the closed caption C2 is assigned to a detailed video V2, and the detailed video V2 thus is increased by the closed caption C2 as much as the other programs.

FIG. 11(b) shows time transition taken place when the program is transferred. First of all, by taking transfer started at Time  $t_4$  as an example, information sufficient to play-back every program to some extent is transferred by Time  $t_{10}$ . It means,

closed caption is transferred to the Programs 1 and 3, and audio is transferred to the Program 2. Note that the program components to be transferred between Time  $t_4$  to Time 10 are the ones having the transfer priority of 1 in the storage list of FIG. 12.

Thereafter, the program components in the "script" class which are not transferred before the lapse of Time  $t_{10}$  are transferred between Time  $t_{10}$  to Time  $t_{11}$ . As being completely transferred by Time  $t_{10}$ , the audio of the Program 2 is not transferred at this stage. The program components to be transferred between Time  $t_{10}$  to Time  $t_{11}$  are the ones having the transfer priority of 2 in the storage list of FIG. 12.

After Time  $t_{11}$ , video is started to be transferred. First of all, the digest video D1 of the Program 1 is transferred. The digest video is equal to the detailed video in the sense but has less information. Having less information means a state, for example, in which on-screen resolution is low or time-axis resolution is low, for example, a frame in a few seconds.

When encoding of the detailed video is hierarchical encoding such as resolution scalability defined in MPEG-2 Video, for example, where videos in a plurality of layers are included in codes, data of the digest video can be obtained by selectively extracting the codes in a particular layer. Further, encoding by a hierarchical transfer method can be done in a similar manner to the hierarchical encoding. Alternately, in a case where a scene being a part on the time-axis of the program is specified

by program index codes adopted in the digital broadcasting system in Japan, only the video in one or more scene(s) specified by the index as more important can be taken out as the digest video. In this case, the remaining scenes do not have video data. However, by displaying the last frame having the video data or by blacking out, it can be regarded as one program component in series.

The videos are transferred between Time  $t_{11}$  and Time  $t_{12}$ . In detail, the detailed video is transferred to the Programs 2 and 3, and the digest video being similar to the detailed video in the sense but has less information is transferred to the Program 1. The videos become available to some extent for every program at Time  $t_{12}$ . The program components to be transferred between Time  $t_{11}$  and Time  $T_{12}$  are the ones having the transfer priority of 3 in the storage list of FIG. 12.

Lastly, the detailed video  $V_1$  of the Program 1 is started to be transferred from Time  $t_{12}$ . For the transfer started from Time  $t_{12}$ , the program components having the transfer priority of 4 in the storage list of FIG. 12 are transferred. It is herein assumed that the detailed video  $V_1$  of the Program 1 was supposed to be completely transferred by Time  $t_{13}$  but was interrupted at Time  $t_b$  exactly at the halfway between Time  $t_{12}$  and Time  $t_{13}$ . If this is the case, as to the program components in the "display" class of the Program 1, the digest video is wholly transferred but the detailed video is half-transferred, only the first-half thereof. Even if data transfer is interrupted due to such

eventuality, the users can enjoy the program with video to some extent because the video information is already provided to every program to some extent. Further, even if the data transfer is interrupted at an earlier stage, the users can enjoy the program with closed caption (additionally, audio).

FIG. 12 is an exemplary storage list to be generated in a case where the received data includes a digest video in the main device 1 of FIG. 1. In FIG. 12, the Program 1 has the digest video in the program component (The third line of the storage list. In the receipt list, the 6th program component to be received). Accordingly, in the processing described in FIG. 4 where transfer priorities are determined (step S206), there is a digest video having the top transfer priority defined for the "display" class. Consequently, the processing of step S207 which re-writes the transfer priorities is not executed. Therefore, the digest video "D" and the detailed video "V", both of which are the types of the program components classified into the "display" class, are set to the transfer priorities by default. Further, as the Program 2 does not have closed caption having the top transfer priority defined for the "script" class, by step S207, the transfer priority of audio "A" is changed into the value of 1 defined as being the top priority for the "script" class.

FIG. 13 is an exemplary receipt list to be generated when the data transferred from the main device 1 to the portable device 2 of FIG. 1 includes a digest video. In FIG. 13, as to the Program

1, the portable device 2 receives the entire digest video of 30 minutes, but receives only the first-half of the detailed video of 15 minutes. In this manner, in step S406, the detailed video reaches to the tail of data 15 minutes after the start of play-back the detailed video and audio in the play-back processing of the program described in FIG. 8 (step S405). Thereafter, the digest video being the program component in the "display" class after the lapse of 15 minutes is found in step S407, then is stated to play-back as a replacement for the detailed video in step S410. The play-back image of the Program 1 is accordingly lowered in quality after the lapse of 15 minutes from the start, however, is still sufficient for the users to enjoy the program.

(Second Embodiment)

FIG. 14 is an outer perspective diagram showing the physical structure of a portable viewing/listening system according to a second embodiment of the present invention. Further, FIG. 15 is a block diagram showing the electrical structure of the portable viewing/listening system according to the second embodiment of the present invention. By referring to FIGS. 14 and 15, the main device 1 includes a processing part 1a, a charging part 1b, and a cradle-type box 1c (hereinafter, refer to as cradle) onto which the portable device 2 can be placed. The portable device 2 includes a processing part 2a, a battery 2b, and a connection contact 2c. The processing part 1a is in a similar structure to the main device 1 shown in FIG. 1, and the processing part 2a is



in a similar structure to the portable device 2 shown in FIG. 1.

By placing the portable device 2 on the cradle 1c, the portable device 2 is electrically connected to the main device 1 via the connection contact 2c. Furthermore, in the meantime, the charging part 1b provided in the main device 1 charges the battery 2b in the portable device 2 through the connection contact 2c. As the portable device 2 is relatively-high power, a rechargeable so-called secondary battery is popular for the battery 2b provided therein. Consequently, the battery 2b needs to be regularly charged every day, and this thus necessitates the portable device 2 to be physically connected to some types of chargers. According to the second embodiment, when any physical connection for the to-be-required charging is made, information such as programs can be advantageously transferred at the same time. Therefore, rather conveniently, there is no need to make any new connection for transfer. Note that, the charging part 1b may monitor charging current and detect the start of connection, and then notifies the detection to the processing part 1a to start the transfer processing of the program. In this manner, rather conveniently, program transfer is automatically started by only placing the portable device 2 on the cradle 1c.

(Third Embodiment)

FIG. 16 is an outer perspective diagram showing the physical structure of a portable viewing/listening system according to a third embodiment of the present invention. By referring to FIG.

16, in the portable viewing/listening system of this embodiment, data transfer from the main device 1 to the portable device 2 is done indirectly via the recording medium 3. To be more specific, the recording medium 3 is written with information such as program when being attached to the main device 1, and is read out the internally-stored program information, for example, when being attached to the portable device 2. In such manner, in the third embodiment, program transfer or information exchange can be done in a similar manner to the first embodiment via the recording medium 3. The recording medium 3 includes a hard disk card or a semiconductor recording card in PCMCIA Standards, for example.

In the third embodiment, the main device 1 is provided with a digital broadcasting receiver equipped with a slot of DVB-CI (Digital Video Broadcasting - Common Interface). The DVB-CI is a specification standardized by the standardization group in Europe called DVB, and specifies the receiver to be equipped with a slot of PCMCIA and the slot to be an insertable type for a descrambler of a PC card in PCMCIA Standards. Accordingly, with a plurality of slots provided and a plurality of descramblers inserted, the receiver can correspond to various cryptographic methods. For the DVB-CI, PCMCIA which is normally equipped with a note-type PC (Personal Computer) as a physical and electrical interface is used. For some receivers, it is structurally possible to function in a similar manner to the main device 1 in FIG. 1 only by updating a software in the existing digital

broadcast receiver by way of download through broadcasting.

In the third embodiment, the portable device 2 is the note-type PC. Thanks to the improving-performance of the CPU, for example, the video and audio of the MPEG-2 can be decoded only through software processing. Therefore, only by installing a computer program which executes the processing for the portable device in the existing-type of the note-type PC, an operation similar to that of the portable device 2 can be attained. Note that the aforementioned computer program is written to the recording medium 3 by the main device 1, for example, and then is read out by the portable device 2 and executed. In such manner, there is no need to have such special operation that the program is downloaded from the network so as to obtain the program in the portable device 2.

FIG. 17 is a block diagram showing the structure of a portable viewing/listening system according to a fourth embodiment of the present invention. In the fourth embodiment, it is assumed that an information source for the main device 1 is not broadcast but broadcast-type service from a computer network 4 such as Internet.

In the fourth embodiment, the main device 1 is provided with a modem 1d, a PC (personal computer) 1e, and an MD (Mini Disk) deck 1f used to exchange data in the computer network 4. The modem 1d, PC 1e, and MD deck 1f are interconnected through a high-speed digital interface such as IEEE1394. In this example, the PC 1e

functions in a similar manner to the main device in FIG. 1 by executing a predetermined computer program. The recording medium 3 may include a package medium such as MD.

In the fourth embodiment, the portable device 2 is structured as a audio-playback-capable digital still camera including an MD drive 2d, a photographing part 2e, an encoding part 2f, a reproduction part 2g, a display 2h, a speaker 2i, and a CPU 2j. Note that the CPU 2j controls an operation of a predetermined circuit block (for example, the MD drive 2d, the photographing part 2e, and the encoding part 2f). By operating the CPU 2j provided for the digital still camera in accordance with the predetermined computer program, the digital still camera is structured to function in a similar manner to the portable device 2 in FIG. 1 in addition to a typical function for a general camera. Note that, the recording medium 3 is not limited to the MD. Further, the portable device 2 may directly receive data from the main device 1 in a structure where the main device 1 is connectable to the portable device 2.

In the embodiments described in the foregoing, every program data stored in the main device 1 is set to be transferred to the portable device 2. However, only the program data selected by the users among the program data stored in the main device 1 may be transferred to the portable device 2. If this is the case, the EPG (Electric Program Guide) data may be previously transferred from the main device 1 to the portable device 2, and

then the users may preselect any program data desired to receive from the main device 1 while taking a look at the program guide at their convenience. In detail, the main device 1 recognizes the programs preselected in the portable device 2 when the portable device 2 is attached thereto, and then transfers only the applicable program data to the portable device 2.

#### INDUSTRIAL APPLICABILITY

The present invention is suitable for a case where a program received by a main device and stored therein is to be easily enjoyed with a portable device.